Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Forward Prediction-based Approach to Target-Tracking with Out-of-Sequence Measurements

François Rhéaume & Abder Rezak Benaskeur
Decision Support Systems Section
Defence R&D Canada – Valcartier

Abstract— **In target-tracking applications, there may be situations where measurements from a given target arrive out of sequence at the processing center. This problem is commonly referred to as Out-of-Sequence Measurements (OOSMs). So far, most of the existing solutions to this problem are based on retrodiction, where backward prediction of the current state is used to incorporate the OOSMs at the appropriate time. This paper suggests a new method for tackling the OOSMs problem without backward prediction. Based on a forward prediction and de-correlation approach, the method has proved to be as performing as the best retrodiction-based methods, while requiring less data storage in most cases.**

I. Introduction

The target-tracking involves a continuous monitoring of the environment in order to collect and process data from the available sources. This tracking process aims at producing a model for the behavior history of the targets of interest [20]. It uses the past measurements of some features of the target to estimate its behavior and predict it in the future. Because of the temporal aspect of the tracking operation, the chronological appearance of the measurements is of prime importance to the estimation process. The measurements are usually tagged with a stamp that indicates the instant at which they were collected.

In many target-tracking applications there is a delay between the moment the measurement is collected from the environment and the moment it is received and processed by the tracking system. Such a delay may vary from one measurement to another, and may arise from many factors, such as sensor diversity, communication delays and unsteady pre-processing times of the observed data. The delay may be long enough so that a measurement taken at a time t_τ reaches the tracking system at a time t_d after the target track has been updated with more recently collected measurements. Such a delayed measurement is referred to as an Out-Of-Sequence Measurement (OOSM) [1]. This problem is also referred to as the problem of tracking with random sampling and delays [14] or negative-time measurement update [1].

A straightforward solution to the OOSM problem is to ignore and discard the OOSM in the tracking process. With such a solution, the information contained in the discarded OOSM is lost. To avoid this, a simple alternative consists of reprocessing, in a chronological order, all measurements collected from the OOSM time t_τ to the last track update time t_k [6]. This solution yields optimal track quality, but remains inefficient because of its high computation and storage requirements. In most cases, it is even unfeasible since most tracking systems keep only the current state estimate and the corresponding error covariance matrix.

Several methods have been proposed to deal with the OOSM problem. Most of these are based on backward prediction of the current state to incorporate the OOSMs, also referred to as the retrodiction approach [1, 2, 5, 6, 8, 10, 14–19, 21]. These methods have to deal with the problem of taking into account the process noise during the retrodiction. The suggested methods in [6, 14, 19] compensate partially for the process noise, or simply ignore it. The major difficulty with retrodiction is that there is a strong dependency between the retrodicted process noise and the current state in backward prediction.

In this paper, a new method is proposed that does not rely on retrodiction, but rather on the forward prediction of the OOSM. A pseudo-track (or tracklet) is created using the OOSM and the track value at a time prior to the OOSM date. Before being fused with the actual track, the created tracklet is predicted forward and de-correlated from the actual track using a track de-correlation method similar to the information filter approach [11]. This method will be referred to as Forward-Prediction Fusion and De-correlation (FPFD). The proposed method has proved to compare favorably to the best retrodiction-based algorithms, while requiring, in most cases, less data storage.

This paper is organized as follows. The problem of OOSMs is stated in Section II. In Section III are discussed the retrodiction-based methods. The proposed FPFD approach is described in Section IV. In Section V, comparison results and performance discussion of the FPFD method are provided.

II. Problem Statement

The dynamical model of the target of interest is assumed to be described by

$$\boldsymbol{x}(k) = \boldsymbol{F}(k, k-1)\boldsymbol{x}(k-1) + \upsilon(k, k-1) \tag{1}$$

with $\boldsymbol{F}(k, k-1)$ being the state transition matrix from time t_{k-1} to time t_k and $\upsilon(k, k-1)$ the effect of the process noise from time t_{k-1} to time t_k. The measurement model is

$$\boldsymbol{z}(k) = \boldsymbol{H}(k)\boldsymbol{x}(k) + \omega(k) \tag{2}$$

where $\boldsymbol{H}(k)$ is the measurement matrix and $\omega(k)$ the measurement noise. The process noise $\upsilon(k, k-1)$ and the measurement noise $\omega(k)$ are assumed to be white with zero mean and covariance matrices $\boldsymbol{Q}(k, k-1)$ and $\boldsymbol{R}(k)$, respectively. It is also assumed that there is no statistical dependency between $\upsilon(k)$ and $\omega(k)$.

The role of the tracking system is to maintain tracks of the targets within the area of interest. A track is composed of a state estimate $\hat{\boldsymbol{x}}$ and the related estimation error covariance matrix $\boldsymbol{P}$. It is assumed that a measurement z is collected and used to update the track after each time interval of length h. At time t_k, the tracking system computes $\hat{\boldsymbol{x}}(k|k)$ and $\boldsymbol{P}(k|k)$

$$\hat{\boldsymbol{x}}(k|k) = E\left[\boldsymbol{x}(k)\middle|Z^k\right] \;\&\; \boldsymbol{P}(k|k) = \mathrm{cov}\left[\boldsymbol{x}(k)\middle|Z^k\right] \quad (3)$$

where Z^k corresponds to the measurement sequence up to time instant t_k, excluding a measurement $z(\tau)$ with a time stamp $t_\tau < t_k$, as shown on Figure 1.

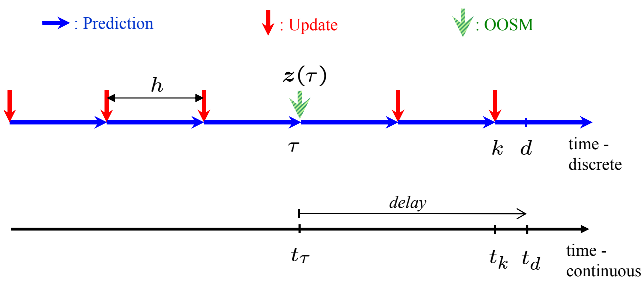

Fig. 1. Out-of-sequence measurement

Once $z(\tau)$ arrives at the processing center, with a certain delay, the problem is to calculate $\hat{\boldsymbol{x}}(k|k,\tau)$ and $\boldsymbol{P}(k|k,\tau)$ that take into consideration the OOSM $z(\tau)$, as follows

$$\hat{\boldsymbol{x}}(k|k,\tau) = E\left[\boldsymbol{x}(k)\middle|Z^k, z(\tau)\right] \quad (4)$$

$$\boldsymbol{P}(k|k,\tau) = \mathrm{cov}\left[\boldsymbol{x}(k)\middle|Z^k, z(\tau)\right] \quad (5)$$

III. Backward prediction methods

A widely used approach to tackle the problem of OOSMs in tracking is based on the "prediction" of the state estimate back to the time of the delayed measurement. Depending on the authors, this approach may be referred to as retrodiction, backward prediction or reverse-time prediction. It sums up to predict the track from the last update time k backward to the time τ of the OOSM, and then update the track using the OOSM and then predict back forward from time instant τ to time instant k.

Various retrodiction methods have been presented in the literature recently. In [1], a retrodiction method is suggested, which is presented as an exact solution compared to the solutions in [6, 14, 19], insofar as the process noise is accounted for entirely during the retrodiction. This solution has proved to be optimal for the case where the OOSM lies between the last two measurements, which is also called the 1-step lag case. The problem with this algorithm, compared to those presented in [14, 19], where the process noise in the retrodiction process is only partially compensated for, is that it requires storage of the innovation. The optimal approach described in [1] and the suboptimal approaches in [14, 19] were extended in [2] to the case of an OOSM with an arbitrary lag, that is an OOSM whose time stamp can be earlier than the last sampling interval. The approach in [2] for an arbitrary lag, which was originally presented in [5], uses an equivalent measurement concept. This concept was first presented in [12, 13] and was shown to provide some advantage over the solutions presented previously in [15, 18, 21]. The equivalent measurement combines together all the measurements that are more recent than the OOSM. This concept was used to extend the algorithms in [1] to the case of an n-step lag OOSM. The authors concluded in [2] that the suboptimal algorithm, which ignores the retrodicted noise, is a good compromise between accuracy and cost.

Still, all of the retrodiction-based methods have some drawbacks. The optimal solutions presented in [18, 21] require a large storage capacity compared to suboptimal OOSM methods. The equivalent measurement concept with backward prediction presented in [2] also necessitates a considerable amount of data storage. Even in the case where the retrodicted noise is ignored, covariance of the equivalent innovation needs to be computed. This requires storage of the error covariance matrices corresponding to the state estimates based on all measurement time stamps that are subsequent to the OOSM. The same applies to the Augmented State Kalman Filter method presented in [8], which also uses equivalent measurements.

IV. FPFD Approach

The proposed Forward-Prediction Fusion and De-correlation (FPFD) method uses a de-correlation technique originally presented in [11]. A forward predicted version of a tracklet that uses the OOSM is fused with the current track using the information form of the Kalman Filter. A tracklet is used, instead of a complete track, to remove the correlation created between the two pieces of information by their common history. Such a de-correlation aims to render the OOSM-based tracklet independent of the process noise for the whole prediction interval.

A. Correlation between the process noise and the state

It is first assumed that there is no update between the time t_τ of the OOSM and the time of the current state t_k, which corresponds to the case of a 1-step lag OOSM. As a consequence, the track from t_τ to t_k is estimated only according to the state prediction model. The corresponding target state can be written as

$$\boldsymbol{x}(k) = \boldsymbol{F}(k,\tau)\boldsymbol{x}(\tau) + \upsilon(k,\tau) \quad (6)$$

where $\boldsymbol{x}(\tau)$ and $\upsilon(k,\tau)$ are not correlated

$$E\left[\boldsymbol{x}(\tau)\upsilon(k,\tau)^T\right] = 0 \quad (7)$$

In the case where there is an update at time $t_\tau < t_{k-1} < t_k$, the state transition from t_τ to t_k is estimated according to the measurement $\boldsymbol{z}(k-1)$ and according to the state prediction model. The corresponding target motion model used by the filter is such that the process noise is cumulated up to the time t_{k-1} of the update, and then cumulated from time t_{k-1} to time t_k. The corresponding target state can be written as

$$\boldsymbol{x}(k) = \boldsymbol{F}(k, k-1)\boldsymbol{x}(k-1) + \upsilon(k, k-1) \quad (8)$$

$$\boldsymbol{x}(k-1) = \boldsymbol{F}(k-1, \tau)\boldsymbol{x}(\tau) + \upsilon(k-1, \tau) \quad (9)$$

where $\boldsymbol{x}(k-1)$ and $\upsilon(k-1, \tau)$ are now correlated:

$$E\left[\boldsymbol{x}(k-1)\upsilon(k-1, \tau)^T\right] \neq 0 \quad (10)$$

and thus $\boldsymbol{x}(k)$ and $\upsilon(k-1, \tau)$ are also correlated:

$$E\left[\boldsymbol{x}(k)\upsilon(k-1, \tau)^T\right] \neq 0 \quad (11)$$

This corresponds to the case of a 2-step lag OOSM. For the n-step lag case and with $n > 1$, we have

$$E\left[\boldsymbol{x}(k)\upsilon(k-i, k-i-1)^T\right] \neq 0, \quad i \in \{1, \ldots, n-1\} \quad (12)$$

Let $\hat{\boldsymbol{x}}(\tau)$ be the state estimate at time t_τ. The forward prediction of $\hat{\boldsymbol{x}}(\tau)$ up to time t_k is

$$\hat{\boldsymbol{x}}(k|\tau) = \boldsymbol{F}(k, \tau)\hat{\boldsymbol{x}}(\tau) \quad (13)$$

When there is no update between $[t_\tau, t_k[$, the forward prediction in (13) is optimal since the process noise $\upsilon(k, \tau)$ for the interval $[t_\tau, t_k[$ is independent of the current state $\boldsymbol{x}(k)$. When there are one or more updates between $[t_\tau, t_k[$, the forward prediction in (13) is not optimal because the current state $\boldsymbol{x}(k)$ is dependent of the process noise $\upsilon(k-1, \tau)$ for the interval $[t_\tau, t_k[$.

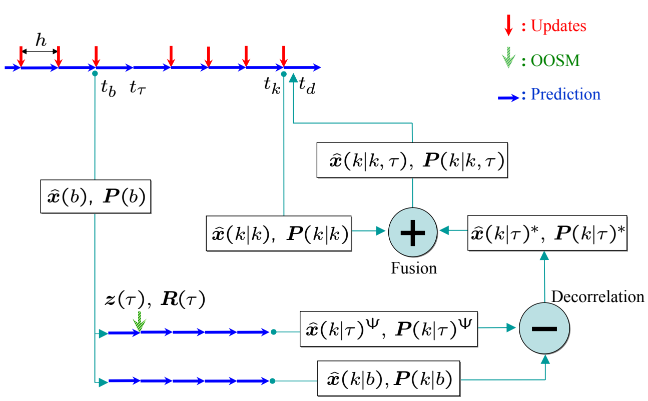

Fig. 2. FPFD method for track update with an OOSM

B. Algorithm

At time k, before the OOSM has been processed, the state estimate and its error covariance matrix are represented by $\hat{\boldsymbol{x}}(k|k)$ and $\boldsymbol{P}(k|k)$ respectively. As shown in Figure 2, in order to incorporate the OOSM $\boldsymbol{z}(\tau)$ into the current state estimate, the OOSM is predicted forwards up to time k. This requires knowledge of the state estimate at a time $t_b < t_\tau$.

The forward predicted OOSM $\boldsymbol{x}(k)^*$ and the corresponding covariance matrix $\boldsymbol{P}(k)^*$ are used to update the track $\hat{\boldsymbol{x}}(k|k), \boldsymbol{P}(k|k)$ at time k. The goal is to have a forward predicted OOSM $\boldsymbol{x}(k)^*$ and its covariance matrix $\boldsymbol{P}(k)^*$ such that the state estimate obtained by using the forward predicted OOSM will be the same as when all the measurements (including the OOSM) are processed sequentially. This requires independence of $\boldsymbol{x}(k)^*$ and $\hat{\boldsymbol{x}}(k|k)$. Consequently, the covariance update for the forward predicted OOSM is given by

$$\boldsymbol{P}^{-1}(k|k)^* = \boldsymbol{P}^{-1}(k|k) + \boldsymbol{P}^{-1}(k)^* = \boldsymbol{P}^{-1}(k|k, \tau) \quad (14)$$

where $\boldsymbol{P}^{-1}(k|k, \tau)$ represents the covariance matrix after the OOSM has been included.

Equation (14) represents the optimal solution, where $\boldsymbol{x}(k)^*$ and $\hat{\boldsymbol{x}}(k|k)$ are independent. In order to have independence, $\boldsymbol{x}(k)^*$ and $\boldsymbol{P}(k)^*$ must be determined such as the dependence between the process noise and the state is totally taken into account in the interval $[t_b, t_k]$. As shown in the previous section, the proposed forward prediction approach is optimal for the 1-step lag case since there is no dependence between the process noise and the state when the OOSM has a 1-step lag. The independence is no longer true for the multiple-step lag cases. In those cases, the proposed algorithm compensates partially for the dependence between the process noise and the current state. The partial independence problem is similar to that of the retrodiction algorithm presented in [2].

Let us consider a forward predicted OOSM $\boldsymbol{x}(k)^\Psi$ and its covariance matrix $\boldsymbol{P}(k)^\Psi$ that depend on the current state $\hat{\boldsymbol{x}}(k|k)$. These are given based on the state estimate $\hat{\boldsymbol{x}}(b)$, with $t_b < t_\tau$, and the corresponding error covariance matrix $\boldsymbol{P}(b)$

$$\boldsymbol{P}(\tau|b)^\Psi = \boldsymbol{F}(\tau, b)\boldsymbol{P}(b)\boldsymbol{F}(\tau, b)^T + \boldsymbol{Q}(\tau, b) \quad (15)$$

$$\boldsymbol{P}^{-1}(\tau|\tau)^\Psi = \boldsymbol{P}^{-1}(\tau|b)^\Psi + \boldsymbol{H}(\tau)^T\boldsymbol{R}^{-1}(\tau)\boldsymbol{H}(\tau) \quad (16)$$

$$\boldsymbol{P}(k)^\Psi = \boldsymbol{F}(k, \tau)\boldsymbol{P}(\tau|\tau)^\Psi\boldsymbol{F}(k, \tau)^T + \boldsymbol{Q}(k, \tau) \quad (17)$$

and

$$\boldsymbol{x}(\tau|b)^\Psi = \boldsymbol{F}(\tau, b)\hat{\boldsymbol{x}}(b) \quad (18)$$

$$\boldsymbol{P}^{-1}(\tau|\tau)^\Psi\boldsymbol{x}(\tau|\tau)^\Psi = \boldsymbol{P}^{-1}(\tau|b)^\Psi\boldsymbol{x}(\tau|b)^\Psi + \boldsymbol{H}(\tau)^T\boldsymbol{R}^{-1}(\tau)\boldsymbol{z}(\tau) \quad (19)$$

$$\boldsymbol{x}(k)^\Psi = \boldsymbol{F}(k, \tau)\boldsymbol{x}(\tau|\tau)^\Psi \quad (20)$$

where $\boldsymbol{F}(j, i)$ represents the state transition matrix from time t_i to time t_j and $\boldsymbol{Q}(j, i)$ is the process noise error covariance matrix for the time interval $[t_i, t_j]$.

$\boldsymbol{x}(k)^\Psi$ and $\hat{\boldsymbol{x}}(k|k)$ are correlated since they share the same history (*i.e.*, $\hat{\boldsymbol{x}}(b)$) and both rely on the same prediction model between t_b to t_k.

The prediction process associates some process noise with both $\boldsymbol{x}(k)^\Psi$ and $\hat{\boldsymbol{x}}(k|k)$ during the interval $[t_b, t_k]$. Such a dependence can be compensated by the de-correlation of $\boldsymbol{x}(k)^\Psi$ from the prediction of the state estimate for the interval $[t_b, t_k]$.

The prediction of the state estimate from time t_b to t_k and the corresponding error covariance matrix are given by

$$\hat{\boldsymbol{x}}_p(k) = \boldsymbol{F}(k,b)\hat{\boldsymbol{x}}(b) \quad (21)$$
$$\boldsymbol{P}_p(k) = \boldsymbol{F}(k,b)\boldsymbol{P}(b)\boldsymbol{F}(k,b)^T + \boldsymbol{Q}(k,b) \quad (22)$$

The de-correlation of the forward predicted OOSM from the current state $\hat{\boldsymbol{x}}(k|k)$ is given by

$$\boldsymbol{P}^{-1}(k)^* = \boldsymbol{P}^{-1}(k)^\Psi - \boldsymbol{P}_p^{-1}(k) \quad (23)$$
$$\boldsymbol{P}^{-1}(k)^*\boldsymbol{x}(k)^* = \boldsymbol{P}^{-1}(k)^\Psi\boldsymbol{x}(k)^\Psi - \boldsymbol{P}_p^{-1}(k)\hat{\boldsymbol{x}}_p(k) \quad (24)$$

where $\boldsymbol{x}(k)^*$ is the resulting de-correlated measurement and $\boldsymbol{P}^{-1}(k)^*$ is its corresponding covariance matrix. Again, the de-correlation of $(\hat{\boldsymbol{x}}(k|k), \boldsymbol{P}(k|k))$ and $(\boldsymbol{x}(k)^\Psi, P(k)^\Psi)$ is only partial for the multi-step lag case.

Finally, the current state estimate $\hat{\boldsymbol{x}}(k|k)$ is updated with the forward predicted OOSM $\boldsymbol{x}(k)^*$. Using the information fusion form as in (14), the covariance update for the translated OOSM is

$$\boldsymbol{P}^{-1}(k|k,\tau) = \boldsymbol{P}^{-1}(k|k) + \boldsymbol{P}^{-1}(k)^* \quad (25)$$

and the state update is

$$\boldsymbol{P}^{-1}(k|k,\tau)\hat{\boldsymbol{x}}(k|k,\tau) = \boldsymbol{P}^{-1}(k|k)\hat{\boldsymbol{x}}(k|k) + \boldsymbol{P}^{-1}(k)^*\boldsymbol{x}(k)^* \quad (26)$$

C. Data storage

Data storage may be of a major importance based on the storage capacity available in a tracking system. Algorithm FPFD is compared to the Al1 and to the Bl1 retrodiction-based algorithms from [2], as well as to the optimal algorithm presented in [21], in terms of storage requirements.

As mentioned in [2], all OOSM algorithms necessitate, at least, the storage of the following data: i) a scalar for the time stamp of the next update; ii) n scalars for the state estimate; and iii) $n(n+1)/2$ scalars for the covariance of the state estimate. Furthermore, algorithm FPFD requires; i) a scalar for the time stamp t_b associated with track $\hat{\boldsymbol{x}}(b), \boldsymbol{P}(b)$; ii) n scalars for the state estimate $\hat{\boldsymbol{x}}(b)$ held in memory at time instant t_b; iii) $n(n+1)/2$ scalars for the covariance $\boldsymbol{P}(b)$ of the state estimate.

Based on the numbers given above, algorithm FPFD requires a storage of n^2+3n+2 scalars. Because the delay of an OOSM cannot be predicted in advance, a maximum number of lags l_{max} needs to be determined. This maximum number of lags l_{max} has a direct impact on the storage requirements of the retrodiction-based algorithms. As shown below, the storage requirements for the FPFD approach do not depend on l_{max}[1]. Details about the storage requirements for algorithms Al1 and Bl1 are provided in [2]. The total number of scalars stored for each OOSM algorithm is given below

$$\textbf{FPFD} \;:\; C_F = n^2 + 3n + 2 \quad (27)$$
$$\text{A}l1 \;:\; C_A = \left\lceil \frac{l_{max}+1}{2} \right\rceil (n^2+3n+2) \quad (28)$$
$$\text{B}l1 \;:\; C_B = \left\lceil \frac{l_{max}+1}{2} \right\rceil (n^2+3n+2) - nl_{max} \quad (29)$$

The data storage requirements of the FPFD, Al1 and Bl1 algorithms are less than those of the optimal algorithm presented in [21], which requires at least[2]

$$\left\lceil \frac{4l_{max}-1}{2} \right\rceil n^2 + \left\lceil \frac{8l_{max}-1}{2} \right\rceil n + l_{max} \quad (30)$$

scalars.

Table I shows the data storage requirements in terms of l_{max} for a state vector of 4 dimensions.

l_{max}	**1**	**2**	**3**	**4**
FPFD	30	30	30	30
Al1	30	45	60	75
Bl1	26	37	48	59
Algorithm I in [21]	39	88	137	186

TABLE I

NUMBER OF SCALARS TO BE STORED

D. Determination of the storage time of the state estimate

Algorithm FPFD requires the storage of the state estimate and its covariance matrix at a time $t_b < t_\tau$. Depending on the context of the tracking application, different approaches can be used to determine the storage time t_b of the state estimate. For example, in the case where the observations' sampling rate is fixed, the estimate can be stored and the memory refreshed at each measurement update until an anticipated measurement update is missed due to a delay (of the OOSM). This would ensure that the estimate is always stored at a time $t_b < t_\tau$ and with $t_b = t_\tau - h$.

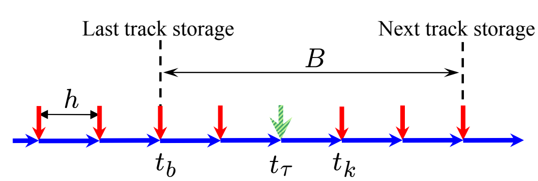

Fig. 3. Case where time t_b is determined according to B. The state estimate and its covariance matrix are held in memory after each time interval B, where $B > h$.

In a more general case, the state estimate could be held in memory after each time interval of length B, where $B > h$. In that case the storage time t_b of the state estimate changes after each time interval of length B, as illustrated in Figure 3. B corresponds to the maximum number of lags l_{max} that are taken into consideration. Note that the state estimate is not stored at the time of the OOSM t_τ.

[1]Although the storage requirements of algorithm FPFD do not depend on a predetermined maximum number of lags l_{max}, its performance in terms of track accuracy does depend on l_{max}. This is discussed in Section IV-D

[2]Data storage requirements for *Case I : Perfect knowledge about τ at time $j+1$* in [21]

The value of B has some influence on the performance of algorithm FPFD. On average, the performance will deteriorate as B increases. More exactly, the performance deteriorates as the difference $t_\tau - t_b$ increases, where we have $\max(t_\tau - t_b) = B$. The deterioration of the performance for increasing $t_\tau - t_b$ differences is due to the correlation between the process noise and the current state, as discussed in Section IV-A.

V. Results and Discussion

To demonstrate the performance of the FPFD method, the latter is compared with the retrodiction-based algorithms presented in [1, 2]. The results reported therein are compared to those yielded by the FPFD method, using the same test scenarios. First, a scenario with a delayed measurement whose time stamp is within the last sampling interval (Example 2 of [1]) is presented. It will be referred to as the 1-step lag scenario. Afterward, we present a n-step lag scenario that is identical to the one used in [2]. This scenario considers the case of delayed measurements whose time stamps are within one or more sampling intervals. Finally, we simulate three OOSM scenarios with 2D nonlinear measurement model and sensor communication delays. The three scenarios are similar to some of the real-world examples presented in [2]. In order to preserve symmetry of the covariance matrices, the Joseph form is used for the covariance update [7]. Also, in every test the storage of the state estimate related to algorithm FPFD is made such that $t_b = t_\tau - h$, except for the results shown in Tables XII and XIII, where different values of the storage interval B are tested.

A. 1-*step lag scenario*

The considered scenario (from [1]) uses a discrete-time dynamical system with three different values for the continuous time process noise variance (q = 0.5, 1, 4). The state equation is given by

$$\boldsymbol{x}(k) = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \boldsymbol{x}(k-1) + \boldsymbol{v}(k, k-1) \tag{31}$$

where h is the sampling interval, and where the process noise $\boldsymbol{v}(k, k-1)$ is assumed white with a zero mean and a covariance matrix

$$E\left[\boldsymbol{v}(k, k-1)\boldsymbol{v}(k, k-1)'\right] = q \begin{bmatrix} h^3/3 & h^2/2 \\ h^2/2 & h \end{bmatrix} = \boldsymbol{Q} \tag{32}$$

The measurement $\boldsymbol{z}(k)$ of the state $\boldsymbol{x}(k)$ is given by

$$\boldsymbol{z}(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}(k) + \boldsymbol{w}(k) \tag{33}$$

where $\boldsymbol{w}(k)$ is white measurement noise, with a zero mean and a covariance matrix

$$E\left[\boldsymbol{w}(k)^2\right] = R = 1 \tag{34}$$

The estimation starts at time $k = 1$, with initial covariance matrix

$$\boldsymbol{P}(1|1) = \begin{bmatrix} R & R/h \\ R/h & 2R/h^2 \end{bmatrix} \tag{35}$$

An OOSM with time stamp $\tau = 1.5$ has to be processed at time $k = 2$.

Based on the above-described scenario, the FPFD method is compared with the in-sequence measurements reprocessing method (In-seq), OOSM discarding, and with the algorithms[3] A, B, and C from [1]. Algorithm A is referred to as the optimal retrodiction algorithm in that it accounts entirely for the process noise. Algorithms B and C are called suboptimal retrodiction algorithms since they ignore the retrodicted noise. Algorithm C is a simpler version of algorithm B. Furthermore, the in-sequence measurement reprocessing method is the simple approach that reprocesses all the past measurements chronologically starting from the OOSM time. It provides the optimal solution[4]). The discard approach comes down to simply ignoring the OOSM.

Performance — As shown in Table II, algorithm A and FPFD are both optimal since they yield the same $\hat{P}(k|k,\tau)$ as the in-sequence measurements reprocessing method, and this for the different values of the process noise. Table III shows that discarding the OOSM has a significant impact on the track quality, since the trace of $\hat{P}(k|k,\tau)$ is 6.1% to 18.4% higher than in the case of the optimal methods.

q	**4**	**1**	**0.5**
In-Seq	$\begin{bmatrix} .6825 & .7396 \\ .7396 & 2.5725 \end{bmatrix}$	$\begin{bmatrix} .6248 & .5018 \\ .5018 & 1.0539 \end{bmatrix}$	$\begin{bmatrix} .6129 & .4526 \\ .4526 & .7626 \end{bmatrix}$
Discard	$\begin{bmatrix} .8636 & .6818 \\ .6818 & 2.5909 \end{bmatrix}$	$\begin{bmatrix} .8421 & .5526 \\ .5526 & 1.0658 \end{bmatrix}$	$\begin{bmatrix} .8378 & .5270 \\ .5270 & .7872 \end{bmatrix}$
FPFD	$\begin{bmatrix} .6825 & .7396 \\ .7396 & 2.5725 \end{bmatrix}$	$\begin{bmatrix} .6248 & .5018 \\ .5018 & 1.0539 \end{bmatrix}$	$\begin{bmatrix} .6129 & .4526 \\ .4526 & .7626 \end{bmatrix}$
A	$\begin{bmatrix} .6825 & .7396 \\ .7396 & 2.5725 \end{bmatrix}$	$\begin{bmatrix} .6248 & .5018 \\ .5018 & 1.0539 \end{bmatrix}$	$\begin{bmatrix} .6129 & .4526 \\ .4526 & .7626 \end{bmatrix}$
B	$\begin{bmatrix} .6826 & .7396 \\ .7396 & 2.5725 \end{bmatrix}$	$\begin{bmatrix} .6249 & .5018 \\ .5018 & 1.0539 \end{bmatrix}$	$\begin{bmatrix} .6129 & .4526 \\ .4526 & .7626 \end{bmatrix}$
C	$\begin{bmatrix} .7143 & .8571 \\ .8571 & 2.3851 \end{bmatrix}$	$\begin{bmatrix} .6364 & .5455 \\ .5455 & 1.0655 \end{bmatrix}$	$\begin{bmatrix} .6190 & .4762 \\ .4762 & .7754 \end{bmatrix}$

TABLE II

Covariance matrices for different process noise values (1-step lag scenario)

For further comparison, the actual Mean Square Error (MSE) was also computed through 10000 Monte Carlo runs. The results are summarized in Table IV, where, as a further demonstration of their optimality, the MSE of algorithm FPFD and that of algorithm A are equal to the MSE yielded by in-sequence measurements reprocessing.

Cost — In terms of track quality, the FPFD method was proved, based on the above results, to be as optimal as method A. Algorithms FPFD and A provided a similar performance, both in terms of MSE and estimation error

[3] Algorithm A is also referred to as the "Y-algorithm" in [9].

[4] Note that algorithm Z_l presented in [16, 21] also provides an optimal solution for the general n-step lag case. However, its storage requirements is high compared to the other sub-optimal OOSM algorithm (see Table I)

q	**4**		**1**		**0.5**	
In-Seq	3.2550		1.6787		1.3754	
Discard	3.4545	(6.1%)	1.9079	(13.7%)	1.6250	(18.4%)
FPFD	3.2550	(0%)	1.6787	(0%)	1.3754	(0%)
A	3.2550	(0%)	1.6787	(0%)	1.3754	(0%)
B	3.2551	(.003%)	1.6787	(0%)	1.3754	(0%)
C	3.0994	(-4.8%)	1.7019	(1.4%)	1.3944	(1.4%)

TABLE III

TRACE OF COVARIANCE MATRICES AND RELATIVE DEVIATION WITH RESPECT TO THE OPTIMAL (1-STEP LAG SCENARIO)

q	**4**	**1**	**0.5**
In-Seq	$\begin{bmatrix} .6895 & .7684 \\ .7684 & 2.5937 \end{bmatrix}$	$\begin{bmatrix} .6274 & .5327 \\ .5327 & 1.1257 \end{bmatrix}$	$\begin{bmatrix} .6192 & .4704 \\ .4704 & .8063 \end{bmatrix}$
Discard	$\begin{bmatrix} .8571 & .6988 \\ .6988 & 2.6211 \end{bmatrix}$	$\begin{bmatrix} .8494 & .5868 \\ .5868 & 1.1389 \end{bmatrix}$	$\begin{bmatrix} .8565 & .5416 \\ .5416 & .8273 \end{bmatrix}$
FPFD	$\begin{bmatrix} .6895 & .7684 \\ .7684 & 2.5937 \end{bmatrix}$	$\begin{bmatrix} .6274 & .5327 \\ .5327 & 1.1257 \end{bmatrix}$	$\begin{bmatrix} .6192 & .4704 \\ .4704 & .8063 \end{bmatrix}$
A	$\begin{bmatrix} .6895 & .7684 \\ .7684 & 2.5937 \end{bmatrix}$	$\begin{bmatrix} .6274 & .5327 \\ .5327 & 1.1257 \end{bmatrix}$	$\begin{bmatrix} .6192 & .4704 \\ .4704 & .8063 \end{bmatrix}$

TABLE IV

MSE FOR 10000 MONTE CARLO RUNS (1-STEP LAG SCENARIO)

covariance estimation. For the cost associated with data storage, and for $l_{max} = 1$, equations (27) to (29) lead to identical requirements for algorithms FPFD and A (*i.e.*, $n^2 + 3n + 2$) and less storage requirement for algorithm B (*i.e.*, $n^2 + 2n + 2$). The counterpart of this lower cost for algorithm B is its lower performance in terms of tracking quality, as shown in Tables II to IV.

B. n-step lag scenario

A dynamic system, identical to the one used in [2], is considered with three different OOSM lags $l = 1, 2, 3$. The system behavior is similar to the 1-step lag case, where the dynamics are given by (31) and (32), except that here both position and velocity are measured. The measurement equation is then

$$\boldsymbol{z}(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}(k) + \boldsymbol{w}(k) \qquad (36)$$

where $\boldsymbol{w}(k)$ has an error covariance matrix

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \qquad (37)$$

The filter is initiated at $t_0 = 0$, with

$$\hat{\boldsymbol{x}}(0|0) = \boldsymbol{z}(0), \quad \boldsymbol{P}(0|0) = \boldsymbol{R} \qquad (38)$$

and ends up at $t_4 = 4s$. The three OOSM lags $l = 1, 2, 3$ correspond to times $\tau = 1.5s, 2.5s$, and $3.5s$, as illustrated in Figure 4.

The in-sequence measurements reprocessing method, algorithm Al1, algorithm[5] Bl1, the OOSM discard solution, and algorithm FPFD are all compared. Algorithms Al1 and Bl1 are the n-step lag extensions to the 1-step lag algorithms A and B presented in [1], respectively. They both use an equivalent measurement concept originally presented in [5].

[5]Bl1 is the one-step equivalent measurement version of the "M-algorithm" defined in [15].

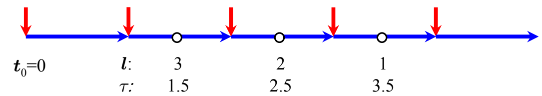

Fig. 4. OOSM with three different lags $l = 1, 2, 3$

Lag	**1**	**2**	**3**
In-Seq	$\begin{bmatrix} .2287 & .0225 \\ .0225 & .0759 \end{bmatrix}$	$\begin{bmatrix} .2597 & .0381 \\ .0381 & .0832 \end{bmatrix}$	$\begin{bmatrix} .2854 & .0387 \\ .0387 & .0833 \end{bmatrix}$
FPFD	$\begin{bmatrix} .2287 & .0225 \\ .0225 & .0759 \end{bmatrix}$	$\begin{bmatrix} .2563 & .0372 \\ .0372 & .0827 \end{bmatrix}$	$\begin{bmatrix} .2906 & .0403 \\ .0403 & .0827 \end{bmatrix}$
Al1	$\begin{bmatrix} .2287 & .0225 \\ .0225 & .0759 \end{bmatrix}$	$\begin{bmatrix} .2563 & .0372 \\ .0372 & .0827 \end{bmatrix}$	$\begin{bmatrix} .2906 & .0403 \\ .0403 & .0827 \end{bmatrix}$
Bl1	$\begin{bmatrix} .2330 & .0254 \\ .0254 & .0779 \end{bmatrix}$	$\begin{bmatrix} .2667 & .0389 \\ .0389 & .0830 \end{bmatrix}$	$\begin{bmatrix} .2955 & .0403 \\ .0403 & .0828 \end{bmatrix}$
Discard		$\begin{bmatrix} .3142 & .0370 \\ .0370 & .0834 \end{bmatrix}$	

TABLE V

COVARIANCE MATRICES FOR DIFFERENT LAG VALUES, $q = 0.5$

Performance — As shown in Tables V and VI, for the n-step lag case ($l > 1$), algorithm FPFD has a performance equal to algorithm Al1. According to the estimated covariance matrix $\hat{P}(k|k,\tau)$, the performance of algorithm FPFD does degrade as the number of step-lag increases, compared to the in-sequence reprocessing of the measurements. The difference is represented in Table VI, where the trace of $\hat{P}(k|k,\tau)$ is presented for the different algorithms, along with the trace´s relative deviation compared to the optimal in-sequence measurements reprocessing. This degradation is due to the dependence issue between the process noise and the state (discussed in Section IV).

Lag	**1**		**2**		**3**	
In-Seq	.3046		.3429		.3687	
Discard	.3976	(30.5%)	.3976	(16.0%)	.3976	(7.8%)
FPFD	.3046	(0%)	.3390	(-1.1%)	.3733	(1.2%)
Al1	.3046	(0%)	.3390	(-1.1%)	.3733	(1.2%)
Bl1	.3109	(2%)	.3497	(2%)	.3783	(2.6%)

TABLE VI

TRACE OF COVARIANCE MATRICES AND RELATIVE DEVIATION WITH RESPECT TO THE OPTIMALITY (n-STEP LAG SCENARIO), $q = 0.5$

Table VII presents the MSE of each algorithm for the n-step lag cases. MSE for algorithms FPFD and Al1 are close to the MSE provided by the in-sequence measurement reprocessing method. Algorithm Bl1 is also relatively close to the optimal method. Note that the performance of algorithms Al1 and Bl1 was discussed thoroughly in [2].

Lag	**2**	**3**
In-Seq	$\begin{bmatrix} .2564 & .0371 \\ .0371 & .0813 \end{bmatrix}$	$\begin{bmatrix} .2835 & .0404 \\ .0404 & .0845 \end{bmatrix}$
Discard	$\begin{bmatrix} .3110 & .0356 \\ .0356 & .0814 \end{bmatrix}$	$\begin{bmatrix} .3120 & .0382 \\ .0382 & .0847 \end{bmatrix}$
FPFD	$\begin{bmatrix} .2592 & .0380 \\ .0380 & .0817 \end{bmatrix}$	$\begin{bmatrix} .2887 & .0408 \\ .0408 & .0848 \end{bmatrix}$
Al1	$\begin{bmatrix} .2592 & .0380 \\ .0380 & .0817 \end{bmatrix}$	$\begin{bmatrix} .2887 & .0408 \\ .0408 & .0848 \end{bmatrix}$
Bl1	$\begin{bmatrix} .2613 & .0375 \\ .0375 & .0815 \end{bmatrix}$	$\begin{bmatrix} .2894 & .0403 \\ .0403 & .0847 \end{bmatrix}$

TABLE VII

MSE FOR DIFFERENT LAG VALUES, $q = 0.5$

Cost — According to equations (27) to (29), the FPFD method has a storage requirement that is equal to the one of algorithm Al1 only for $l_{max} = 1$ (1-step lag situation). For n-step lag scenarios, where $l_{max} > 1$, algorithm FPFD requires less storage space than algorithm Al1. This is also the case with algorithm Bl1. As shown previously, algorithm Bl1 requires less storage for the 1-step lag case ($l_{max} = 1$). However, its storage requirements are larger with the n-step lag scenarios ($l_{max} > 1$).

C. 2D nonlinear measurement model and sensor communication delays

The following example is drawn from the practical examples in [2, 16]. It aims at comparing the FPFD method against other OOSM algorithms when nonlinear measurements conversion are considered. As in [2, 16], a target is tracked using two GMTI sensors. The target motion follows the constant velocity model in two dimensions with process noise spectral density $q = 1m^2/s^3$. The two GMTI sensors have nearly orthogonal lines-of-sight and both have a slant range of about $100km$ from the target. Each sensor observation is in polar coordinates and includes range (r), azimuth (θ) and range rate ($\dot{r}$). The related standard deviations are $10m, 1mrad$ and $1m/s$ respectively. Since the bias significance factor is below 0.4 ($r\sigma_\theta^2/\sigma_r \approx 0.001$), the measurements are converted to Cartesian coordinates using the conventional coordinate transformation [3, 6]. Three scenarios are simulated. Scenario 1-SL, 3-SL and 5-SL see sensor 1 have its last measurement delayed with one lag, three lags and five lags respectively. The lists of the measurements sent to the central tracker by the two GMTI sensors are presented in Table VIII for the three scenarios. The initial target state is [70000m 70000m 60m/s 20m/s]. Track initialization is made according to the two-point initialization technique [4].

Table IX shows the trace of the average filter covariance matrix and the trace of the actual MSE matrix for the OOSM algorithms Al1 and FPFD. Clearly, for the three scenarios, the results are the same whether algorithm Al1 or algorithm

	Trace($\mathbf{P}$)			Trace(MSE)		
Scenario	1l	3l	5l	1l	3l	5l
FPFD	252.39	249.24	253.62	251.63	259.68	260.03
Al1	252.39	249.24	253.62	251.63	259.68	260.03

TABLE IX

TRACES OF COVARIANCE AND MSE MATRICES FOR Al1 AND FPFD

Scenario	1l	3l	5l
In-Seq	3.87	3.97	4.01
FPFD	3.87	4.00	4.14
Al1	3.87	4.00	4.14
Bl1	3.87	4.01	4.08

TABLE X

NORMALIZED ESTIMATION ERROR SQUARED (NEES) AT LAST UPDATE TIME

FPFD is used. Table X shows the normalized estimation error squared (NEES) [4] for the 4-dimensional state based on 1000 runs. Both algorithms Al1 and FPFD result in a NEES of 3.87 in scenario 1l, 4.00 in scenario 3l and 4.14 in scenario 5l. The NEES for algorithms Al1, Bl1 and FPFD all lie within the two-sided 95% confidence bounds based on the χ^2_{4000} distribution (3.8261, 4.1767) [4]. Therefore, algorithms Al1, Bl1 and FPFD are statistically consistent for the three scenarios.

The aim of this practical example is to show that algorithm FPFD has the same performance as algorithm Al1 in a practical OOSM example that involves nonlinear measurement conversions. The results presented in Tables IX and X are conclusive, since the measured performance of algorithm FPFD is identical to the measured performance of algorithm Al1.

Moreover, Table XI shows the CPU times of algorithms Al1, Bl1 and FPFD for 1000 Monte Carlo runs. Although the measured CPU times represent only imprecise approximations of the computational complexity of the algorithms, they are used here for comparison purpose. The measured CPU times of algorithm FPFD are comparable to those of algorithm Bl1, which are lower than those of algorithm Al1 in all of the three lag cases shown in Table XI.

Lag	**1**	**3**	**5**
FPFD	0.71	0.67	0.73
Al1	0.82	0.84	0.82
Bl1	0.75	0.68	0.69

TABLE XI

CPU TIMES (s) FOR 1000 MONTE CARLO RUNS.

B	**1**	**4**	**6**
FPFD	3.99	4.01	4.02
In-Seq		3.99	

TABLE XII

NEES FOR FPFD AT LAST UPDATE TIME, FOR THE 1-STEP LAG OOSM

Scenario 1-SL	Sensor ID	1	2	1	2	1	2	1	2	1	2	2	1
	Time Stamp (s)	0	2.5	5	7.5	10	12.5	15	17.5	20	22.5	27.5	25
Scenario 3-SL	Sensor ID	1	2	1	2	1	2	1	2	2	1	2	1
	Time Stamp (s)	0	2.5	5	7.5	10	12.5	15	17.5	22.5	25	27.5	20
Scenario 5-SL	Sensor ID	1	2	1	2	1	2	2	1	2	1	2	1
	Time Stamp (s)	0	2.5	5	7.5	10	12.5	17.5	20	22.5	25	27.5	15

TABLE VIII

COMMUNICATION DELAYS IN A 2 GMTI RADAR NETWORK, FOR 1, 3, 5-STEP LAG OOSM SCENARIOS

	Trace($\mathbf{P}$)			Trace(MSE)		
B	1	4	6	1	4	6
FPFD	252.3	250.8	250.4	247.1	247.4	247.9
In-Seq		252.3			247.1	

TABLE XIII

TRACES OF COVARIANCE AND MSE MATRICES FOR FPFD

Finally, Tables XII and XIII show the performance of algorithm FPFD according to the storage interval B for the 1-step lag scenario described in Table VIII (Scenario 1-SL). The NEES obtained with algorithm FPFD and shown in Table XII are 3.99, 4.01 and 4.02 for $B = 1$, $B = 4$ and $B = 6$ respectively. Therefore, the NEES increases slightly as B augments. Recall that the related two-sided 95% confidence bounds for 1000 Monte Carlo are $[3.8261, 4.1767]$. The traces of the actual MSE matrices shown in Table XIII also increase slightly as B augments, while the traces of the average filter covariance matrices decrease as B augments. Note that for $B > 1$ algorithm FPFD loses its optimality compared to the in-sequence measurements reprocessing method.

VI. CONCLUSIONS

A forward prediction and decorrelation-based method for processing OOSMs was presented. In terms of track quality, the proposed method was proved to be optimal for the 1-step lag case. For the multiple-step lag case, the method loses its optimality compared to the in-sequence measurements reprocessing. Nonetheless, its results are valuable since they are equal to those obtained with some of the most recent retrodiction methods presented in the literature, while requiring less data storage. Finally, the performance of the FPFD method depends on the storage time of the track. However, when the sampling rate is fixed so that it is known when a measurement has not arrived, the time of track storage can be brought close to the OOSM time. In such conditions, the FPFD method can represent a good choice for practical applications.

REFERENCES

[1] Y. Bar-Shalom, "Update with out-of-sequence measurements in tracking: Exact solution," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-38, no. 3, pp. 769–778, 2002.

[2] Y. Bar-Shalom, H. Chen, and M. Mallick, "One-step solution for the multistep out-of-sequence measurement problem in tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-40, no. 1, pp. 27–37, 2004.

[3] Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. Storrs, CT: YBS Publishing, 1995.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, Wiley, Ed. New York: Wiley, 2001.

[5] Y. Bar-Shalom, M. Mallick, H. Chen, and R. Washburn, "One-step solution for the general out-of-sequence measurement problem in tracking." in *Proceedings of the 2002 IEEE Aerospace Conference*, Big Sky, MT, March 2002.

[6] S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[7] R. Bucy and P. Joseph, *Filtering for Stochastic Processes, with Applications to Guidance*, N. York:, Ed. Wiley, 1968.

[8] S. Challa, R. J. Evans, and X. Wang, "Track-to-track fusion using out-of-sequence tracks," in *Proceedings of the Fifth International Conference on Information Fusion*, Annapolis, Maryland, jul 2002.

[9] ——, "A bayesian solution to oosm problem and its approximations," *Information Fusion*, vol. 4, no. 3, Sept. 2003.

[10] S. Challa, R. J. Evans, X. Wang, and J. Legg, "A fixed lag smoothing framework for oosm problems," *Communications in Information and Systems*, vol. 2, no. 4, pp. 327–350, 2002.

[11] C. Chee-Yee, S. Mori, and K. Chang, "Information fusion in distributed sensor networks," in *American Control Conference*, 1985, pp. 830–835.

[12] J. M. Covino and B. J. Griffiths, "A new estimation method for multisensor data fusion," in *Proceedings of SPIE Conference on Sensor and Sensor Systems for Guidance and Navigation*, vol. 1478, 1991, pp. 114–125.

[13] O. Drummond, "Track fusion with feedback," in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, vol. 2759, 1996, pp. 342–360.

[14] R. D. Hilton, D. A. Martin, and W. D. Blair, "Tracking with time-delayed data in multisensor systems," Dahlgren, VA, Tech. Rep. NSWCDD/TR-93/351, aug 1993.

[15] M. Mallick, S. Coraluppi, and C. Carthel, "Advances in asynchronous and decentralized estimation," in *Proceedings of 2001 IEEE Aerospace Conference*, mar 2001.

[16] M. Mallick, K. Zhang, and X. Li, "Comparative analysis of multiple-lag out-of-sequence measurement filtering algorithms," in *Proc. Signal and Data Processing of Small Targets*, San Diego, CA, August 4-7 2003.

[17] S. Maskell, R. Everitt, R. Wright, and M. Briers, "Multi-target out-of-sequence data association," in *Proceedings of 7th International Conference on Information Fusion*, 2004.

[18] E. W. Nettleton and H. Durrant-Whyte, "Delayed and asequent data in decentralized sensing networks," in *Proceedings of SPIE Conference*, vol. 4571, oct 2001.

[19] S. C. A. Thomopoulos and L. Zhang, "Decentralized filtering with random sampling and delay." *Inf. Sci.*, vol. 81, no. 1-2, pp. 117–137, 1994.

[20] E. Waltz and J. Llinas, *Multisensor data fusion*. Artech House. Inc, 1990.

[21] K. S. Zhang, X. R. Li, and Y. M. Zhu, "Optimal update with out-of-sequence observations for distributed filtering," in *Proceedings of the Fifth International Conference on Information Fusion*, jul 2002.